МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение
высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНОЙ МАТЕМАТИКИ И ПРОГРАММИРОВАНИЯ

vk.com/id446425943
vk.com/club152685050

КУРСОВАЯ РАБОТА
ЗАЩИЩЕНА С ОЦЕНКОЙ

РУКОВОДИТЕЛЬ

Асс.		А.А.Фоменкова
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К КУРСОВОЙ РАБОТЕ

ПОСТРОЕНИЕ ГРАФИКА ФУНКЦИИ ПРИ РЕШЕНИИ ЧИСЛЕННЫХ ЗАДАЧ В MATLAB

по дисциплине: ИНФОРМАТИКА

Санкт-Петербург 2018

СОДЕРЖАНИЕ

ВВЕДЕНИЕ....3
1 ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ....4
2 ОПИСАНИЕ ФУНКЦИИ В MATLAB....6
3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ....8
3.1 Блок схема алгоритма....8
3.2 Описание функционала программы....10
4 ТЕСТИРОВАНИЕ....12
4.1 Основное меню....12
4.3 Работа с пользовательскими значениями....13
4.4 Справочная информация о программе....14
4.5 Обработка ошибок....15
ВЫВОДЫ....17
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ....18
ПРИЛОЖЕНИЕ....19

vk.com/id446425943
vk.com/club152685050

ВВЕДЕНИЕ

Анализ данных удобно начинать с их графического изображения. График позволяет сразу получить общее представление обо всей совокупности показателей и придают изложению данных большую наглядность, чем таблицы, выразительность, облегчают их восприятие и анализ. График позволяет зрительно оценить характер изучаемого явления, присущие ему закономерности, тенденции развития, взаимосвязи с другими показателями.

При помощи графического изображения данных решаются многие задачи исследования: наглядное представление величины показателей (явлений) в сравнении друг с другом; характеристика структуры какого-либо явления; изменение явления во времени; зависимость изменения одного явления от изменения другого; выделение точек экстремума функции.

В данной курсовой работе рассматривается задача построения графика пользователя при решении численных задач в MATLAB. Построение графиков в MATLAB производится посредством нанесения на систему координат заданных точек и их соединений прямыми. Так, чем меньше расстояние между точками, тем более «гладким» кажется график при отображении.

1 ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Вариант 2. Интерполяция таблично заданной функции методом spline интерполяции

Таблица 2. Зависимость динамического коэффициента вязкости ацетона от температуры, мПа (сП)*

Температура	0 °C	10 °C	20 °C	30 °C	40 °C	60 °C	80 °C	100°C
Динамический коэффициент вязкости	0,395	0,356	0,322	0,293	0,268	0,23	0,2	0,17
Проверочная точка 50 °C - 0,246 мПа								

Необходимо написать программу, которая для таблично заданной пользователем функции рассчитает значение функции в точке, не заданной в таблице. Значения функции задаются в порядке возрастания аргумента. Аргумент функции задается с точностью до 2 знаков после запятой.

Программа должна работать следующим образом:

При запуске программы консоль и рабочая область MATLAB должны быть очищены. В консоль должна быть выведена короткая информация о программе (что реализует программа) и главное меню программы. По запросу пользователя должно реализовываться одно из действий:

1. Демонстрация работы программы. Данные для демонстрации работы программы представлены в таблице 2. При демонстрации работы программы должно быть выведено значение в проверочной точке и графическое представление результатов работы программы.

2. Расчет значения функции для произвольного значения аргумента, заданного пользователем. Программа должна запросить у пользователя сначала все значения аргумента, а после – соответствующие им значения функции. Если количество введенных пользователем значений функции и аргумента не совпадают выводится сообщение об ошибке и предлагается повторить ввод или вернуться в главное меню. Далее программа должна запросить значение аргумента, для которого необходимо рассчитать значение функции. Если введенное значение не входит в заданный диапазон значений аргумента, выводится сообщение об ошибке и предлагается повторить ввод (*или вернуться в главное меню*). После расчета программа должна запросить у пользователя – рассчитать другую точку или выйти в главное меню.4 * Введенные пользователем значения необходимо проверить на корректность ввода, т.е. введенные значения должны быть числовыми*.

3. Графическое представление зависимости, введенной пользователем. Если ранее не был выполнен ввод функции, программа сообщает об ошибке и предлагает ввести функцию. При графическом представлении зависимости, введенной пользователем, должно быть выведено одно графическое окно, в котором синей сплошной линией толщиной 2 должна быть прорисована интерполяционная кривая, зелеными ромбами – исходные точки. На графике красной звездочкой должна быть отмечена точка, в которой

определялось значение функции. Если пользователь задавал несколько точек, на графике прорисовать последнюю. График и оси на нем должны быть подписаны.

4. Вывод в консоль информации о программе и авторе.

5. Завершение работы программы Данные (номера пунктов меню), введенные пользователем, должны проверяться на корректность. При некорректном вводе пункта меню должно выводиться сообщение об ошибке и пользователю должно быть предложено повторить ввод.

Обработку данных необходимо провести методом spline интерполяции с помощью встроенной функции MATLAB (функция interp1).

Информация о программе и авторе должна содержать: Название программы, ее назначение, используемый метод решения, дата последней редакции, ФИО, группу разработчика, контакты.

После окончания выполнения выбранного пользователем пункта меню пользователю должно быть предложено вернуться в главное меню или завершить выполнение программы с помощью клавиши «D». При завершении работы программы данные из рабочей области и консоли должны быть удалены, все графики закрыты.

2 ОПИСАНИЕ ФУНКЦИИ В MATLAB

MATLAB обладает мощным инструментом визуализации результатов вычислений. В простейшем случае для построения графиков используется функция `plot`.

Функция `plot` в наиболее простом случае может иметь один аргумент – вектор или матрицу. В таком случае график будет построен в зависимости от номера элемента. Функция `plot` может принимать 2 аргумента: два вектора одинаковой длины, задающие точки для построения графика. Первый аргумент – это координаты точек по оси абсцисс, а второй – соответствующие координаты по оси ординат. Задание аргументами функции `plot` нескольких пар векторов одинаковой длины позволяет построить несколько графиков в одном графическом окне.

В данной курсовой работе рассматривается построение форматированного графика одной функции. В таком случае функция `plot` имеет следующий синтаксис:

```
plot(x, y, LineSpace, 'Name','Value')
```

Выходных параметров функция не имеет.

Входные параметры:

`x` – вектор аргументов функции, double;

`y` – вектор значений функции, double;

`LineSpace` – строковая переменная, содержащая набор параметров, задающих цвет, тип линии и маркеры графика, char;

`Name` - имя дополнительного параметра форматирования графика, выбирается из перечня спецификаций MATLAB и записывается в кавычки;

`Value` - значение дополнительного параметра форматирования графика, тип данных завивит от соответствующего дополнительного параметра форматирования.

Вектора x и y должны быть одинаковой длины.

Параметры, определяющие форматирование графика, записываются в одинарных кавычках, то есть переменная `LineSpace` представляет собой строку. Тип линии, маркера и цвет могут следовать в строке в любой последовательности. По умолчанию, если не задано форматирование линий, график выводится сплошной линией без маркеров. Цвет линий выводится из первых шести, начиная с синего.

Пример использования функции `plot`

```
>> x=0:0.1:10;
>> y=sin(x)./x;
>> plot(x,y,'r--','LineWidth',2)
```

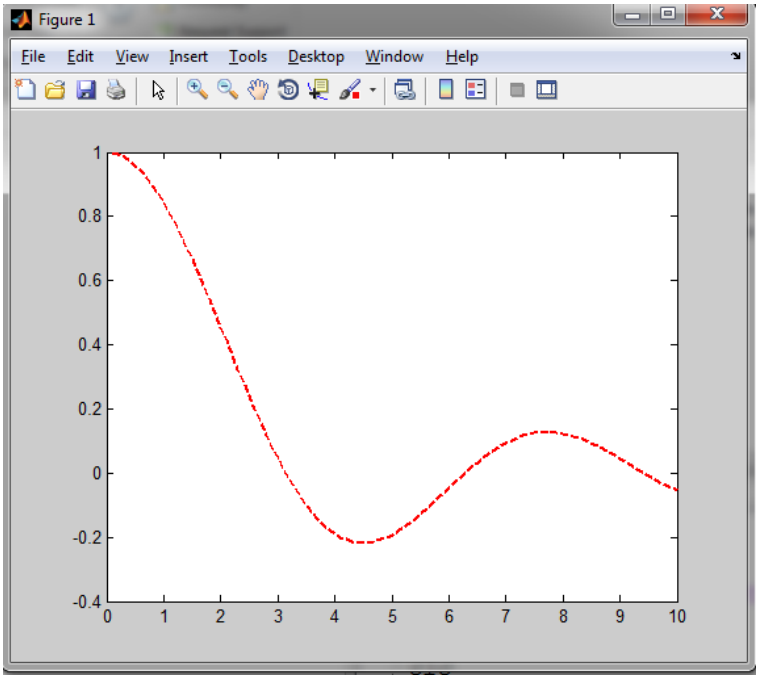

Рисунок 1. Пример построения графика с помощью функции `plot`

3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

3.1 Блок схема алгоритма

Для построение графика пользователя, был написан скрипт `simpleDialog`, организациющий диалог с пользователем. Из скрипта `simpleDialog` вызываются подпрограммы, реализующие отдельные задачи данной курсовой работы.

Работа программы соответствует сценарию главного меню, приведенному на рис. 2.

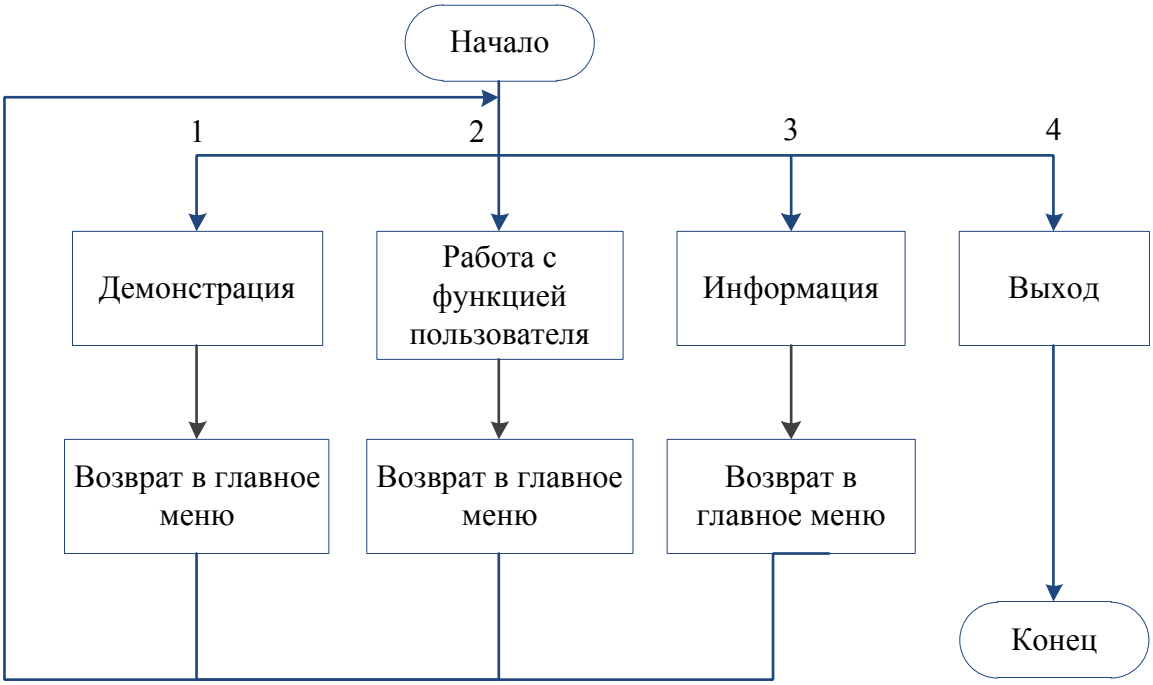

Рисунок 2. Сценарий главного меню

На рисунке 3 представлена блок-схема алгоритма работы скрипта `simpleDialog`.

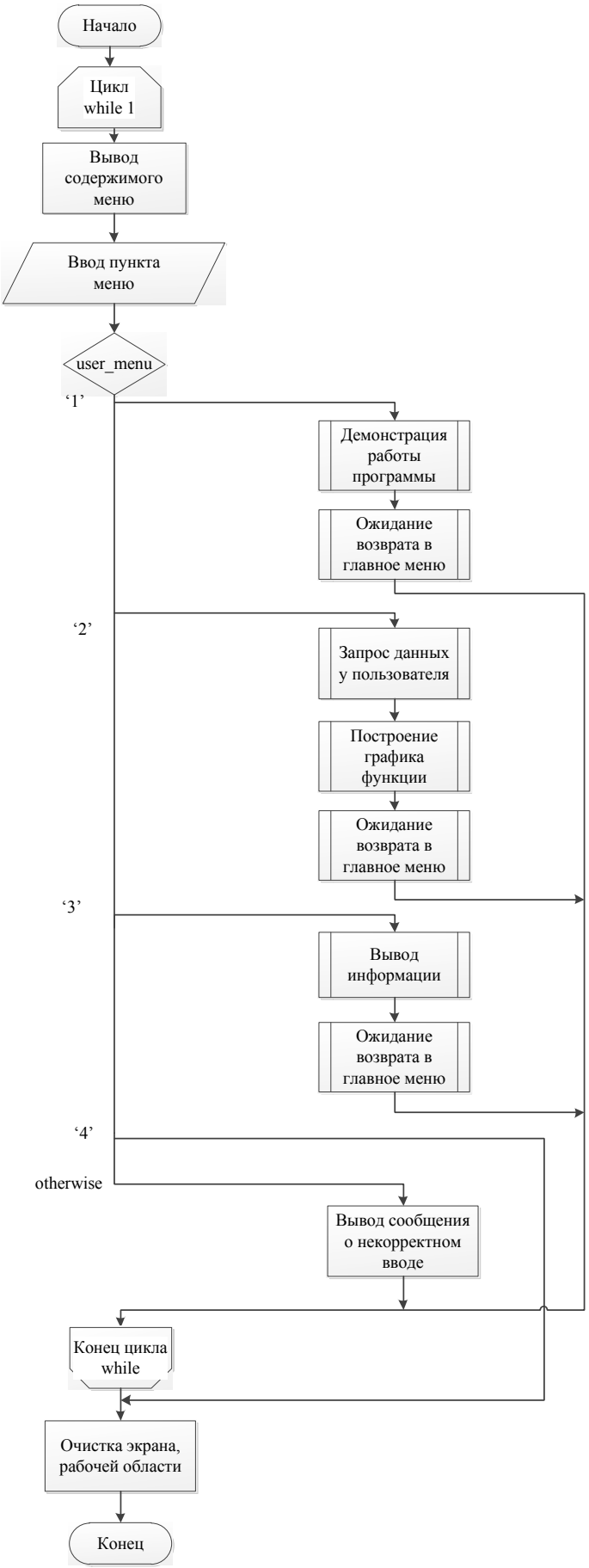

Рисунок 3. Блок-схема алгоритма работы программы `simpleDialog`

3.2 Описание функционала программы

Задачей представленной программы является построение графика функции пользователя с помощью встроенной функции в MATLAB. Программа позволяет задать точки аргумента и функции пользователю самостоятельно с клавиатуры. В другом случае, если пользователю требуется просмотреть пример работы программы без ввода данных, в демонстрационном примере программы имеются заданные по умолчанию параметры аргумента и функции. В программе производится проверка на корректность данных, введенных пользователем. Также пользователь имеет возможность ознакомится с общей информацией о работе программы, ее разработчике.

Для реализации описанного выше функционала программы были разработаны несколько скриптов и функция, описание которых дано в таблицах 1 и 2.

Таблица 1. Перечень функций, использованных в программе `simpleDialog`

Имя функции	**Входные параметры**	**Выходные параметры**	**Описание функционала**
`demonstration`	-	-	Функция демонстрирует работу программы с исходными данными «по умолчанию»
`menu_in`	-	-	Функция осуществляет выход в основное меню по нажатию любой клавиши. При выходе закрываются окна всех графиков, очищается экран
`zapros_dannih`	-	x – аргумент функции, числовой вектор y – значение функции для аргумента x, числовой вектор	Функция запрашивает у пользователя и ее аргумент, рассчитывает значение функции
`Graphik`	x – аргумент функции, числовой вектор y – значение функции для аргумента x, числовой вектор		Функция строит форматированный график y(x)
`inform`	-	-	Функция выводит на экран информацию о программе и авторе

Таблица 2. Перечень вспомогательных функций, использованных в функции `zapros_dannih`

Имя функции	Входные параметры	Выходные параметры	Описание функционала
`more_less`	-	xmin – левая граница интервала, скаляр xmax – правая граница интервала, скаляр	Функция запрашивает два числа - границы интервала, на котором будет рассчитана некоторая функция. Введенные пользователем границы интервала проверяются на то, что они являются числами, на то, что левая граница меньше правой.
`check_number()`	str – строка, содержащая приглашение к вводу	x – корректно введенное число, скаляр	Функция запрашивает у пользователя ввести число, проверяет, является ли введенное данное числовым, если нет, выводит сообщение об ошибке и предлагает повторить ввод до тех пор, пока не будет введено число (скаляр)

4 ТЕСТИРОВАНИЕ

4.1 Основное меню

При запуске программы на экран выводится основное меню (рис.4).

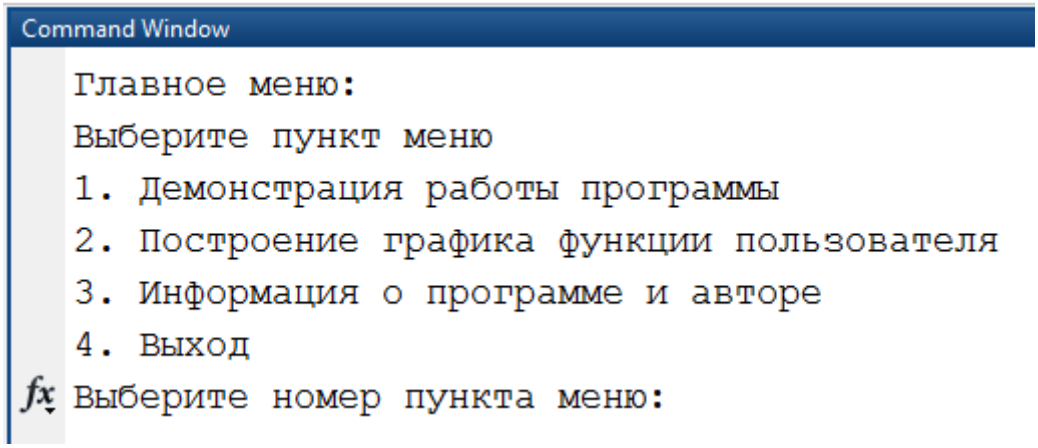

Рисунок 4. Основное меню программы

Пользователю предлагается ввести номер выбранного пункта. В зависимости от введенного значения выполняются определенные действия.

4.2 Демонстрационная работа

При выборе первого пункта меню запускается демонстрационный режим программы. На рис. 5 показан график демонстрационной функции.

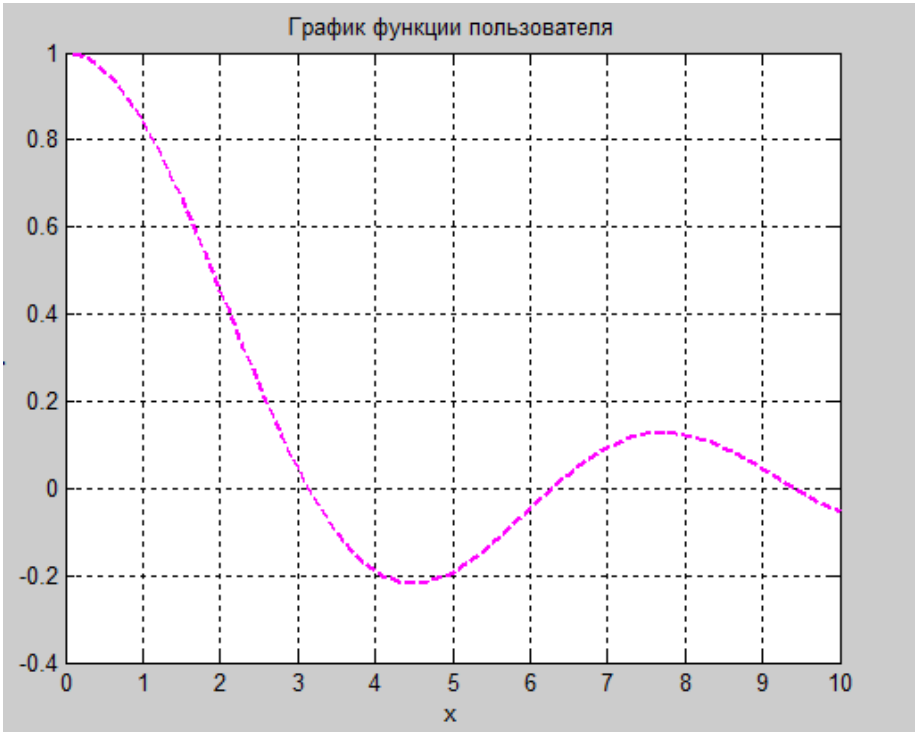

Рисунок 5. Вывод демонстрационного графика

После выполнения демонстрации пользователю будет предложено вернуться в основное меню.

```
Для возврата к меню нажмите любую клавишу:
```

Рисунок 6. Ожидание возврата в главное меню

4.3 Работа с пользовательскими значениями

При выборе второго пункта меню пользователю предлагается ввести с клавиатуры: функцию y(x), минимальное и максимальное значение аргумента и шаг дискретизации (рис. 7).

```
Главное меню:
Выберите пункт меню
1. Демонстрация работы программы
2. Построение графика функции пользователя
3. Информация о программе и авторе
4. Выход
Выберите номер пункта меню: 2
Введите функцию с аргументом x:
y(x) = sin(x)

Введите левую границу интервала:
xmin=0

Введите правую границу интервала:
xmax=5

Задайте шаг:
dx=0.1
Для возврата к меню нажмите любую клавишу:
```

Рисунок 7. Ввод данных пользователем в консоль

В результате на экран будет выведен график (рис.8)

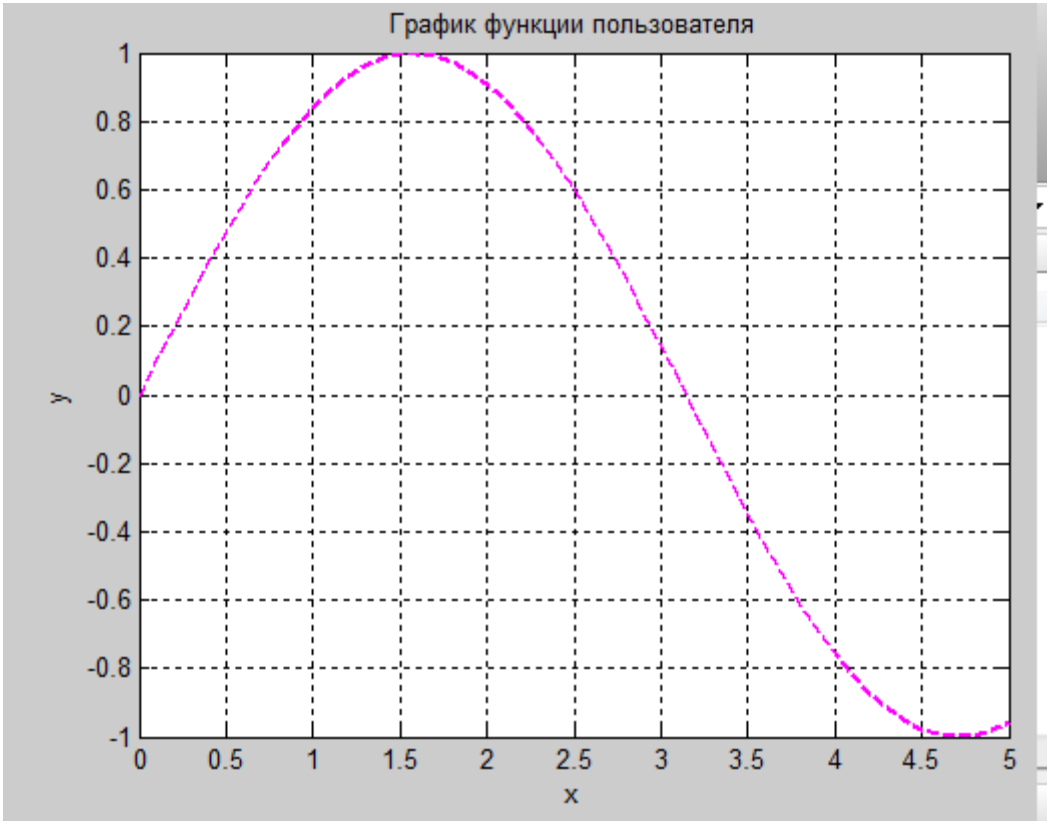

Рисунок 8. График по заданным пользователем точкам

4.4 Справочная информация о программе

При выборе третьего пункта меню, в командном окне будет выведена справочная информация (рис. 9) о названии программы, ее назначении, используемом методе решения, дате последней редакции, ФИО, группе разработчика, контактах.

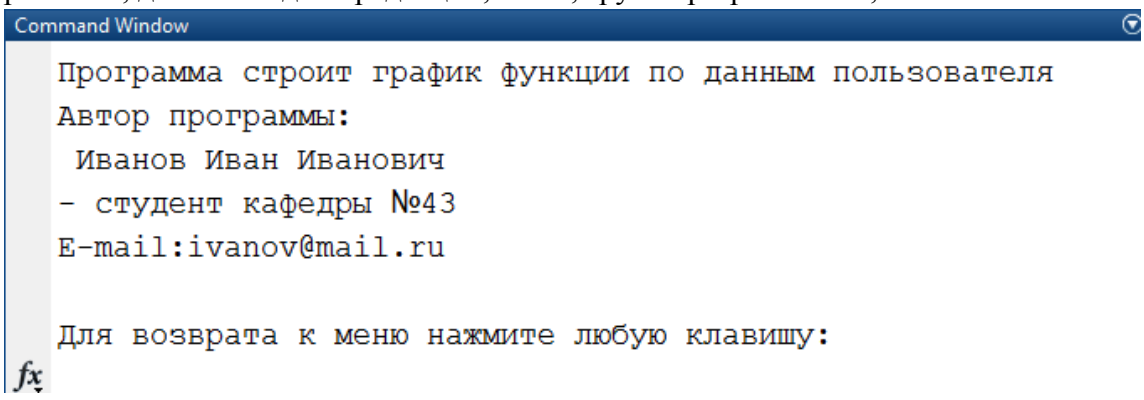

Рисунок 9. Справочная информация

При выборе 4 пункта меню программа будет завершена, графики закрыты, консоль и рабочая область очищены

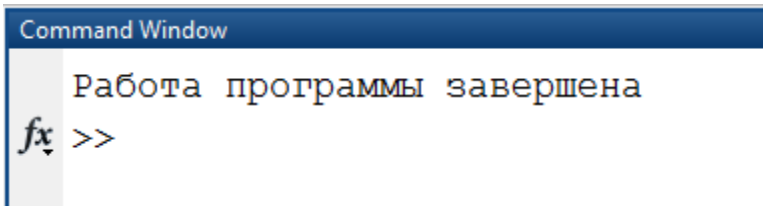

Рисунок 10. Завершение работы программы

4.5 Обработка ошибок

В программе производится проверка данных пользователя на корректность ввода.
При некорректном вводе пункта меню программа укажет на ошибку и предложит повторить ввод (рис.11).

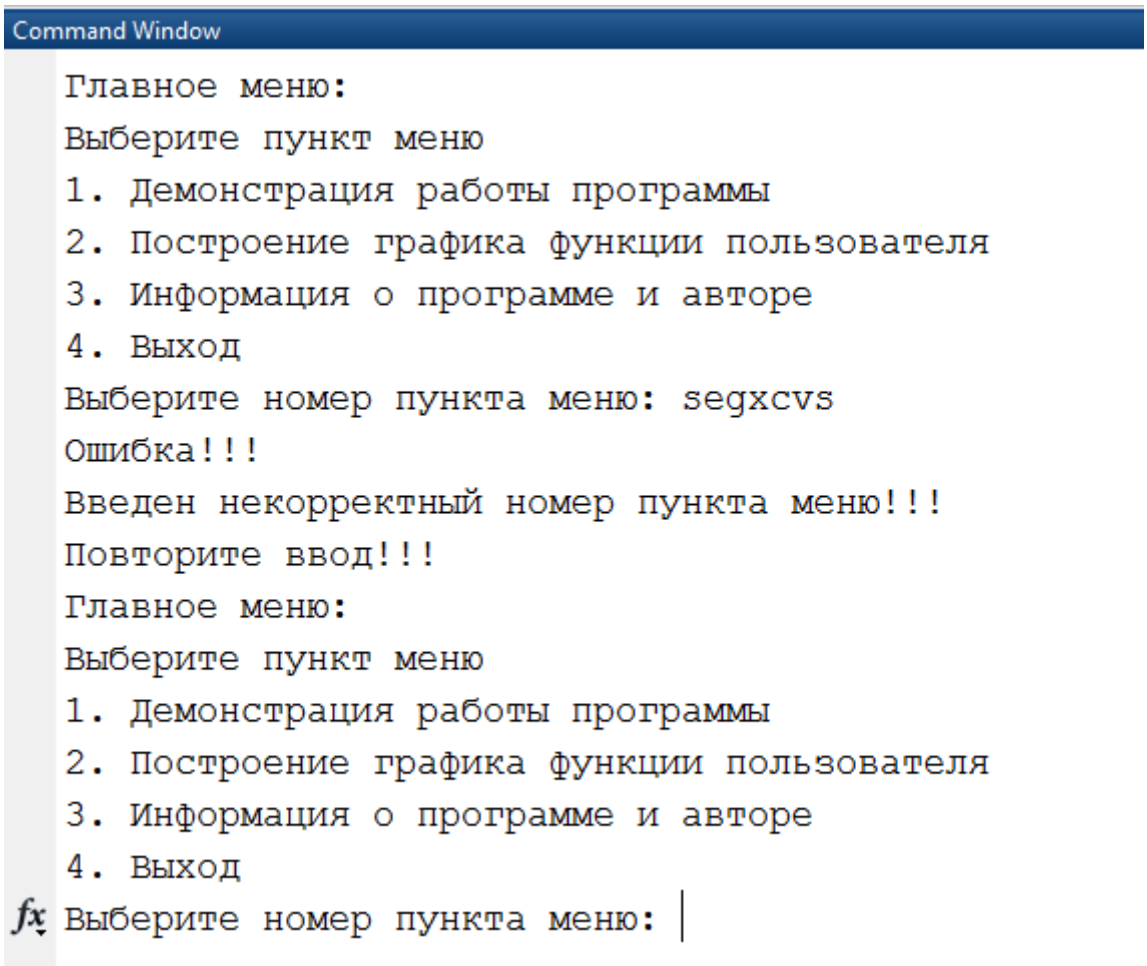

Рисунок 11. Вывод ошибки о некорректном вводе пункта меню

Так же при вводе границ интервала для построения графика функции пользователя программа предлагает повторить ввод, если значение xmin больше значения xmax (рис. 12)

```
Введите левую границу интервала:
xmin=5

Введите правую границу интервала:
xmax=0

Ошибка!!!
Левая граница больше или равна правой
повторите ввод


Введите левую границу интервала:
xmin=
```

Рисунок 12. Вывод ошибки о некорректном вводе границ построения графика

Если введенное значение окажется не скаляром, программа также предложит повторить ввод (рис. 13).

```
Введите левую границу интервала:
xmin=sdfg

Ошибка!!!
Введенное данное содержит символы или пробел!
Повторите ввод

Введите левую границу интервала:
xmin=
```

Рисунок 13. Вывод ошибки о некорректном вводе числа

ВЫВОДЫ

В результате выполнения курсового проекта я разобрался в особенностях функции, реализующей построение графиков функции в MATLAB.

Разработана программа на языке MATLAB, включающая в себя реализацию пользовательского интерфейса в виде диалогового меню для построения графика функции пользователя. В программе производится проверка вводимых данных на корректность и вывод полученного решения в виде текста и графиков.

Разработанная программа протестирована во всех режимах работы. Результаты тестирования подтверждают правильность работы программы.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Амосов А.Л., Дубинский Ю.Л., Копченова Н.В. Вычислительные методы для инженеров: Учеб. пособие. — М.: Высш. шк., 1994. — 544 с: ил.

2. Дьяконов В. П. MATLAB 7.*/R2006/R2007: Самоучитель. – М.: ДМК Пресс, 2008. – 768 с.: ил.

ПРИЛОЖЕНИЕ

simpleDialog.m:

```
% Пример организации более сложного диалога с пользователем
% Очистим рабочую область и консоль
clear all; clc;

% Организация диалога
% Организуем цикл для вывода содержимого меню
% Выход из цикла будем осуществлять командой break

while 1
    disp('Главное меню:');
    disp('Выберите пункт меню');
    disp('1. Демонстрация работы программы');
    disp('2. Построение графика функции пользователя');
    disp('3. Информация о программе и авторе');
    disp('4. Выход');
    % Просим пользователя ввести пункт меню,
    % принимаем как строковую переменную
    user_menu = input('Выберите номер пункта меню: ','s');

    switch user_menu
        case '1'
            % При выборе 1 пункта меню выводится
            % демонстрационный график:
            demonstration
            % Возвращаемся в главное меню по нажатию любой
            % клавиши
            menu_in
        case '2'
            % Запрашиваем у пользователя функцию
            [x,y]=zapros_dannih;
            % строим график функции
            Graphik(x,y)
```

```
            menu_in
        case '3'
            % Выводим на экран информацию о программе и авторе
            inform
            menu_in
        case '4'
            % при выборе 4 пункта меню завершаем выполнение
            % программы
            break

        otherwise
            % под otherwise попадает любое некорректно введенное
            % значение
            disp('Ошибка!!!')
            disp('Введен некорректный номер пункта меню!!!')
            disp('Повторите ввод!!!')
    end
end
% Очищаем консоль, рабочую область, закрываем графики
clc; clear; close;
disp('Работа программы завершена')
```

```
function demonstration
% Функция, демонстрирующая работу программы с исходными данными
% «по умолчанию»

% Вводим данные для построения графика
x = 0 : 0.1 : 10;
y = sin(x) ./ x;
% Строим график
Graphik(x,y)
end
```

```
function Graphik(x,y)
```

```
plot(x,y,'--m','LineWidth',2)
title('График функции пользователя');
% Подписываем оси
xlabel('x');
ylabel('y');
% Включаем сетку
grid on
```

Inform

```
clc
disp('Программа строит график функции по данным пользователя')
fprintf('Автор программы:\n Иванов Иван Иванович\n- студент
кафедры №43\n');
fprintf('E-mail:ivanov@mail.ru\n\n');
```

```
function [x,y] = zapros_dannih
% Функция запрашивает у пользователя функцию и ее аргумент,
% рассчитывает значение функции
% Входных параметров функция не имеет
% Выходные параметры:
% x - значение аргумента, числовой вектор
% y - значение функции, числовой вектор

% Запрашиваем у пользователя функцию как строку
fBody = input('Введите функцию с аргументом x:\ny(x) = ','s');
% Запрашиваем значения аргумента, используем функцию
% more_less
[xmin,xmax] = more_less;
% Запрашиваем шаг и проверяем на корректность его ввода
% Создаем переменную dx
dx = xmax-xmin;
```

```
while (xmax-xmin)<=dx
    dx = check_number('\nЗадайте шаг:\ndx=');
    % Если шаг больше разницы границ интервала, просим
    % пользователя повторить ввод
    if (xmax-xmin)<=dx
        disp('Шаг задан некорректно!!!')
        disp('Повторите ввод!!!')
    end
end
% Создаем вектор аргумента
x = xmin:dx:xmax;
% Рассчитываем значения функции
y = eval(fBody);
end
```

```
function [xmin,xmax] = more_less
%% Функция запрашивает два числа - границы интервала, на котором
будет рассчитана некоторая функция
% Введенные пользователем границы интервала проверяются на то,
что они
% являются числами, на то, что левая граница меньше правой
%% Входные данные отсутствуют
%  Выходные данные -
% xmin – левая граница интервала, скаляр
% xmax – правая граница интервала

%% Функция вызывает функцию check_number для проверки ввода
чисел

while true
    xmin = check_number('\nВведите левую границу интервала:\
nxmin=');
    xmax = check_number('\nВведите правую границу интервала:\
nxmax=');
    if xmin>=xmax
```

```
        disp(' ')
        disp('Ошибка!!!')
        disp('Левая граница больше или равна правой')
        disp('повторите ввод')
        disp(' ')
    else
        break
    end
end
```

```
function menu_in
% выход в основное меню по нажатию любой клавиши
% при выходе закрываются окна всех графиков, очищается консоль

disp('Для возврата к меню нажмите любую клавишу:');
pause
close; clc;
end
```

```
function x = check_number(fBody)
%% Функция запрашивает у пользователя ввести число, проверяет,
является ли
% введенное данное числовым, если нет, выводит сообщение об
ошибке и
% предлагает повторить ввод до тех пор, пока не будет введено
число
% (скаляр)

%% Входные данные
% fBody - строка, содержащая запрос для ввода
% Выходные данные
% x - числовая переменная
while 1
        buffer = input(fBody,'s');
         x = str2double( buffer );  % функция str2double
```

```
                % преобразует строку в число (скаляр) типа
double.
                % Если строка содержит не число, то результатом
                % преобразования будет Nan. При преобразовании
                % символ ',' просто опускается.
        if isnan(x)
                 disp(' ')
                 disp('Ошибка!!!')
                 disp('Введенное данное содержит символы или
пробел!')
                 disp('Повторите ввод')
        elseif length(strfind( buffer,',' ))>0 % в данном случае
              % функция strfind возвращает вектор,
              % элементами которого являются порядковые номера
              %ячеек строки buffer, где содержится подстрока
','.
              % Если подстрока ',' не найдена, функция вернет
              % пустой вектор, длина которого равна 0
                disp(' ')
                disp('Ошибка!!!')
                disp('Введенное данное содержит запятую!')
                disp('Для разделения целой и дробной части числа
используйте точку и повторите ввод!')
        else
                break
        end
end
end
```